High Performance Computing for Information Retrieval

Giuseppe Totaro, PhD
Scientific Applications Software Engineer (Level 2)
Computer Science for Data Intensive Applications (398M)

Jet Propulsion Laboratory
California Institute of Technology

Section

- 3980 - INSTRUMENT SOFTWARE AND SCIENCE DATA SYSTEMS

Information Retrieval

> *"Information retrieval (IR) is finding material (usually documents) of an **unstructured nature** (usually text) that satisfies an **information** need from within **large collections** (usually stored on computers)."*
>
> [Manning, Raghavan and Schutze (2008)]

- IR systems by the scale
  - *web search*
  - *personal information retrieval*
  - *domain-specific search*
- Challenging blend of science and engineering
- First major concept in IR: the ***inverted index***

Inverted Index

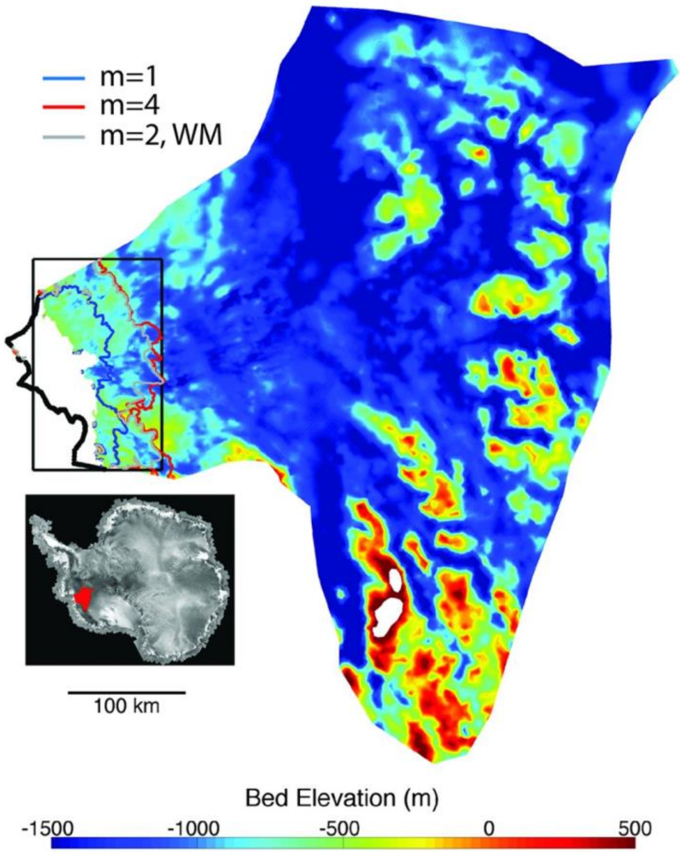
1. Collect the documents to be indexed
2. Tokenize the text
3. Produce a list of normalized tokens
4. Index the documents

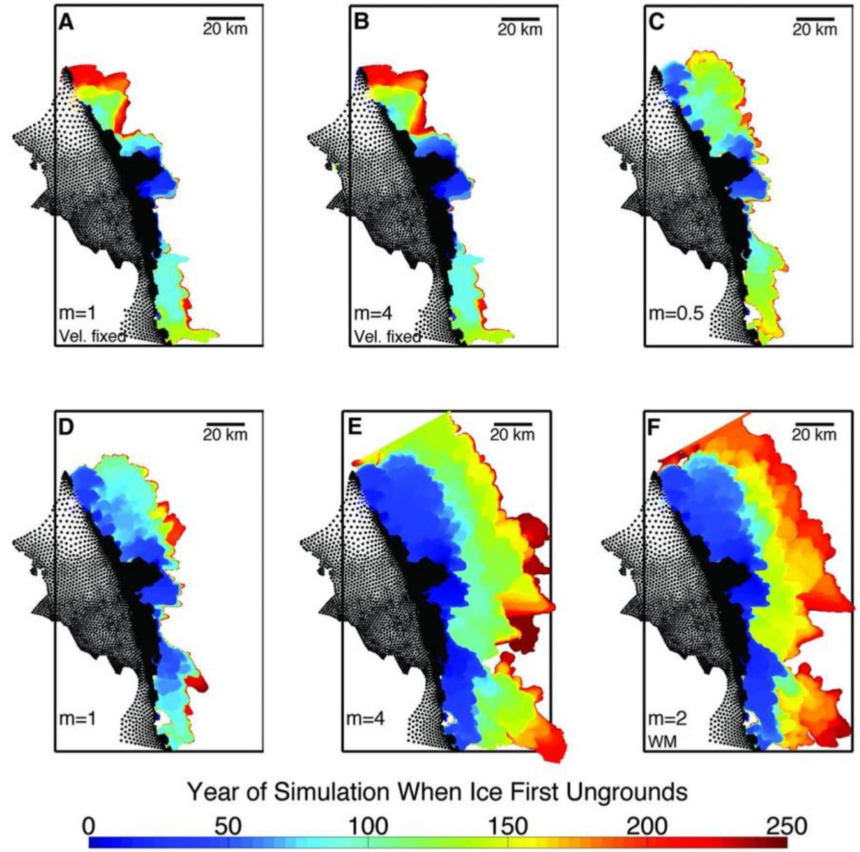
DICTIONARY — POSTINGS

term	doc. freq.	documents
forecasts	1	1:1, 4:6
home	4	1:5, 2:1, 3:13, 4:10
in	2	2:17, 3:10-24
increase	1	3:1
july	3	2:20, 3:27, 4:1
new	2	1:1, 2:6
rise	2	2:12, 4:21
sales	4	1:10, 2:6, 3:18, 4:15
top	1	1:16

Scientific Case Study

Automatic Classification and Interpretation of Polar Datasets



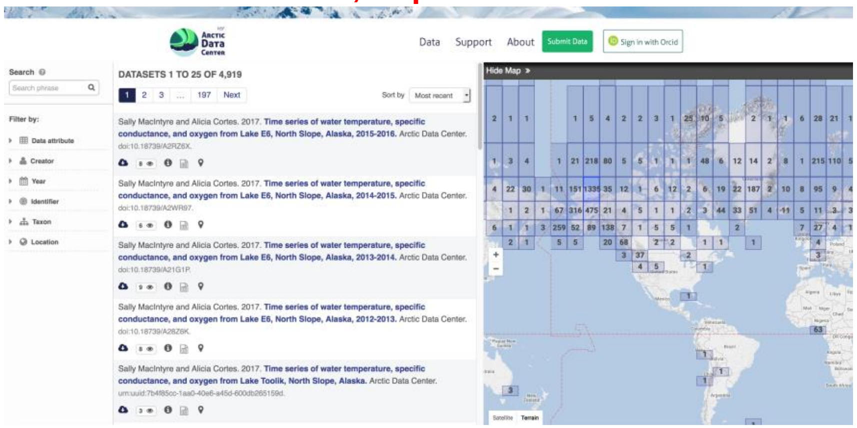


Thwaites Glacier bed topography derived from airborne radar data. [Joughin, Smith and Medley (2014)]

Scientific Case Study

Automatic Classification and Interpretation of Polar Datasets

NSF ADC, https://arcticdata.io



NSIDC ADE, http://arctic-data-explorer.labs.nsidc.org

NASA AMD, https://gcmd.gsfc.nasa.gov

[Burgess and Mattmann (2014)]

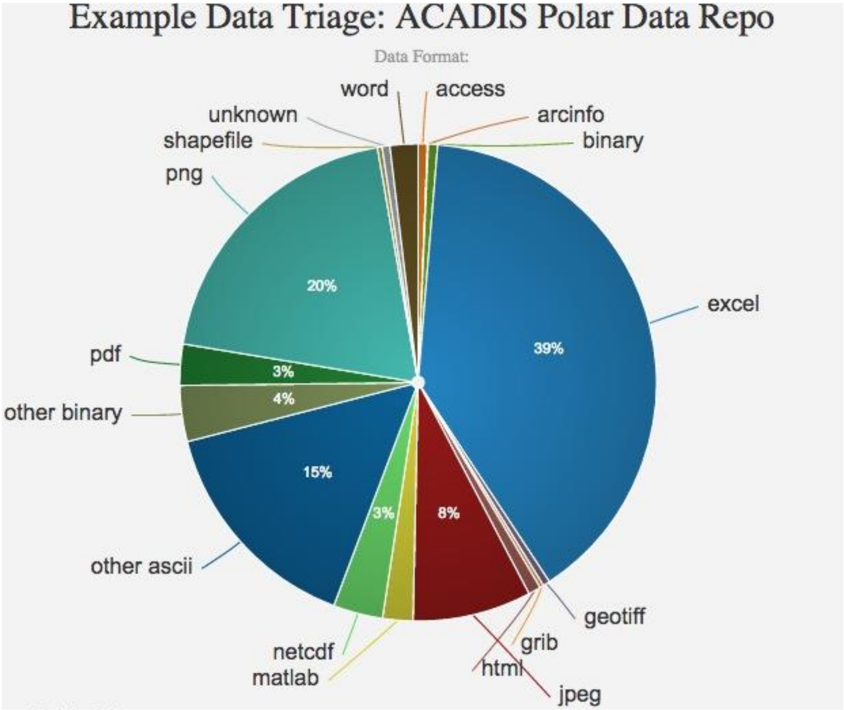
Scientific Case Study

Automatic Classification and Interpretation of Polar Datasets

- TREC Dynamic Domain: Polar Science
  - Data Triage
  - Web Crawl
  - Data Preparation
  - Dataset Characteristics
  - Querying the Data

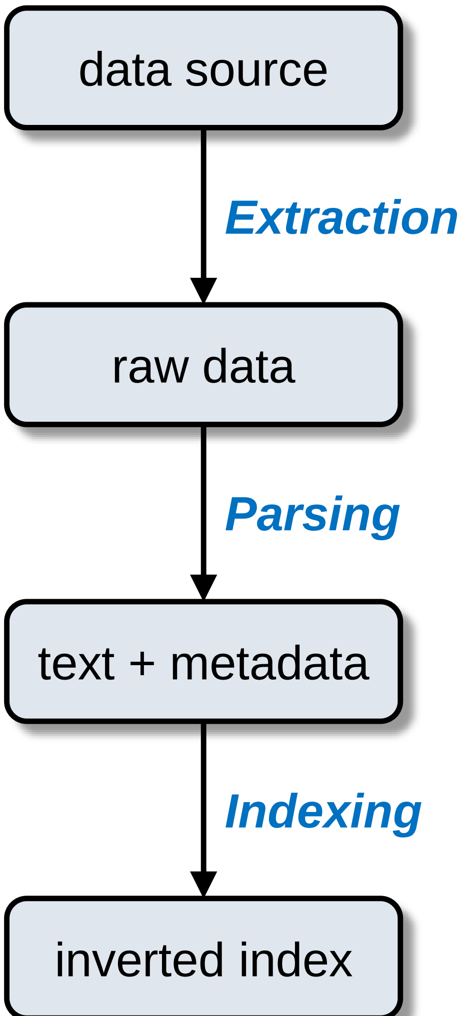
http://trec-dd.org/2015/2015.html

Dataset of 50,000+ **crawled web pages**, **scientific data (HDF, NetCDF files, Grib files)**, zip files, PDFs, images and science code related to the polar sciences and available publicly from the NSF ADC, NASA AMD, and NSIDC ADE.



[Burgess, Mattmann, Totaro, McGibbney and Ramirez (2014)]

Indexing Data Flow



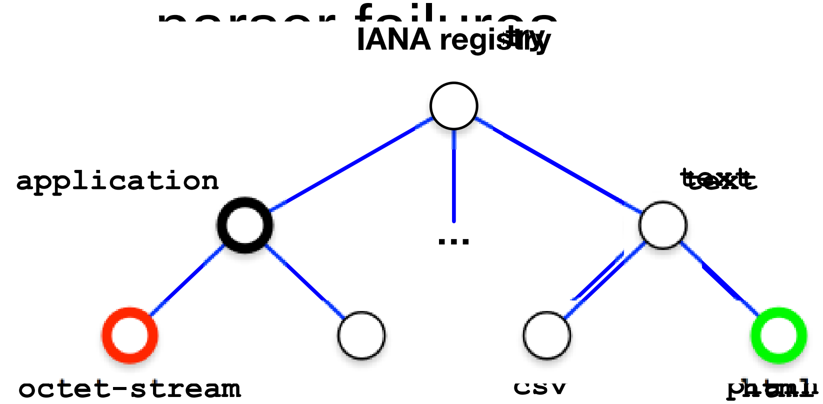
Extracting raw files from data sources (e.g., HDF, NetCDF, GRIB, PDF, etc)

Parsing operations:
- mimetype detection
- language detection
- content handling
- metadata extraction

Indexing operations:
- tokenization
- stop words removal
- stemming

- Main bottlenecks
  - parsing of documents
  - text analysis
  - merging of segments

- Detection complexity



Overview of the Research Problem

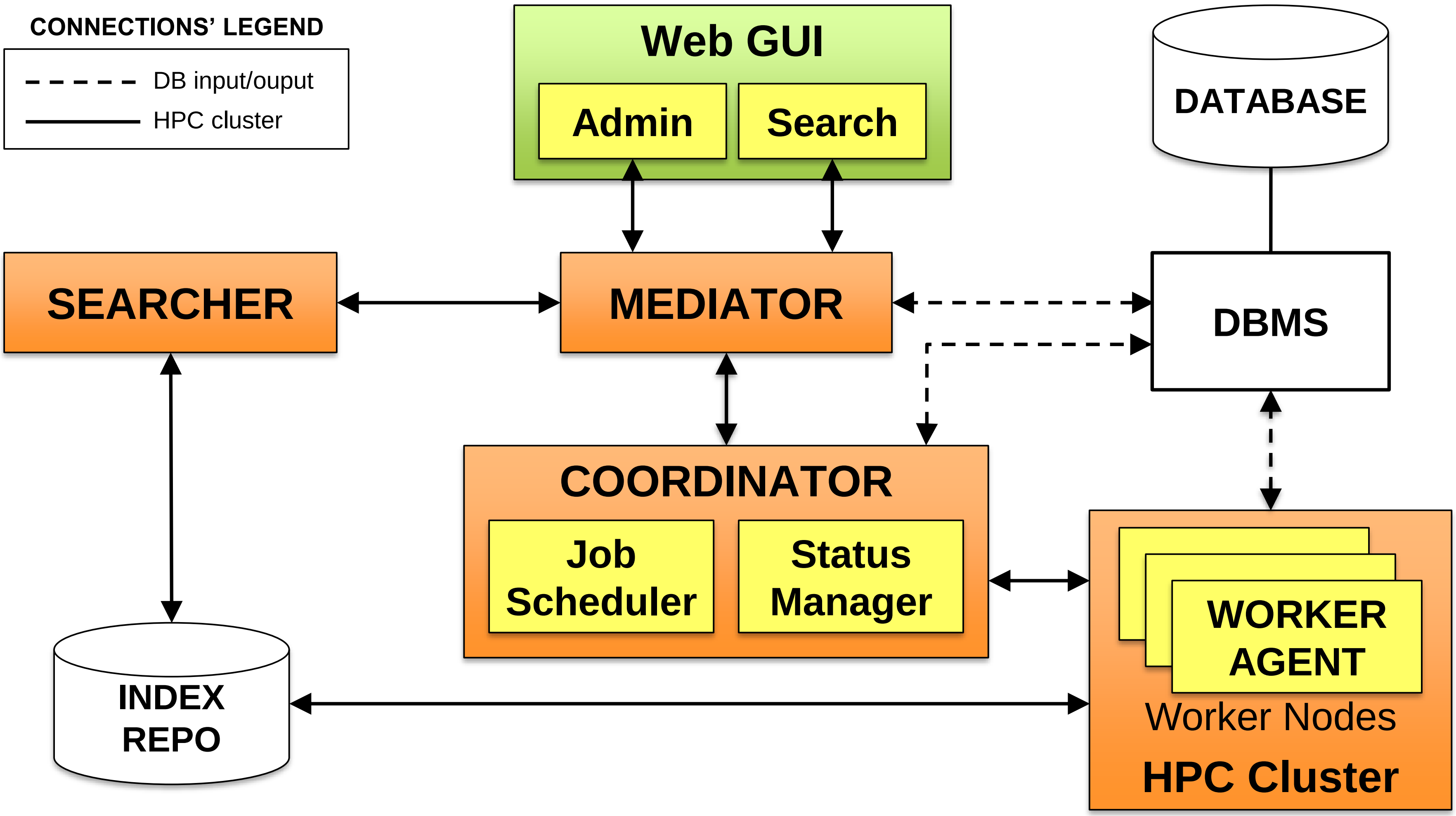
- Fast **indexing** and **searching** of Big Data
  - *inverted index* construction is a burdensome operation
- What is the ***Hadoop ecosystem***?
  - *MapReduce* [Dean and Ghemawat (2010)]
  - *HDFS* [Shafer, Rixner and Cox (2010)]
  - Ecosystem projects: *Hive*, *Spark*, *Ambari*, *Pig*, etc.
- Why not Hadoop?
  - Hadoop performance
  - Data Not Distributed

[Jiang et al. (2010)] [Lin et al. (2012)] [Dong et al. (2014)]

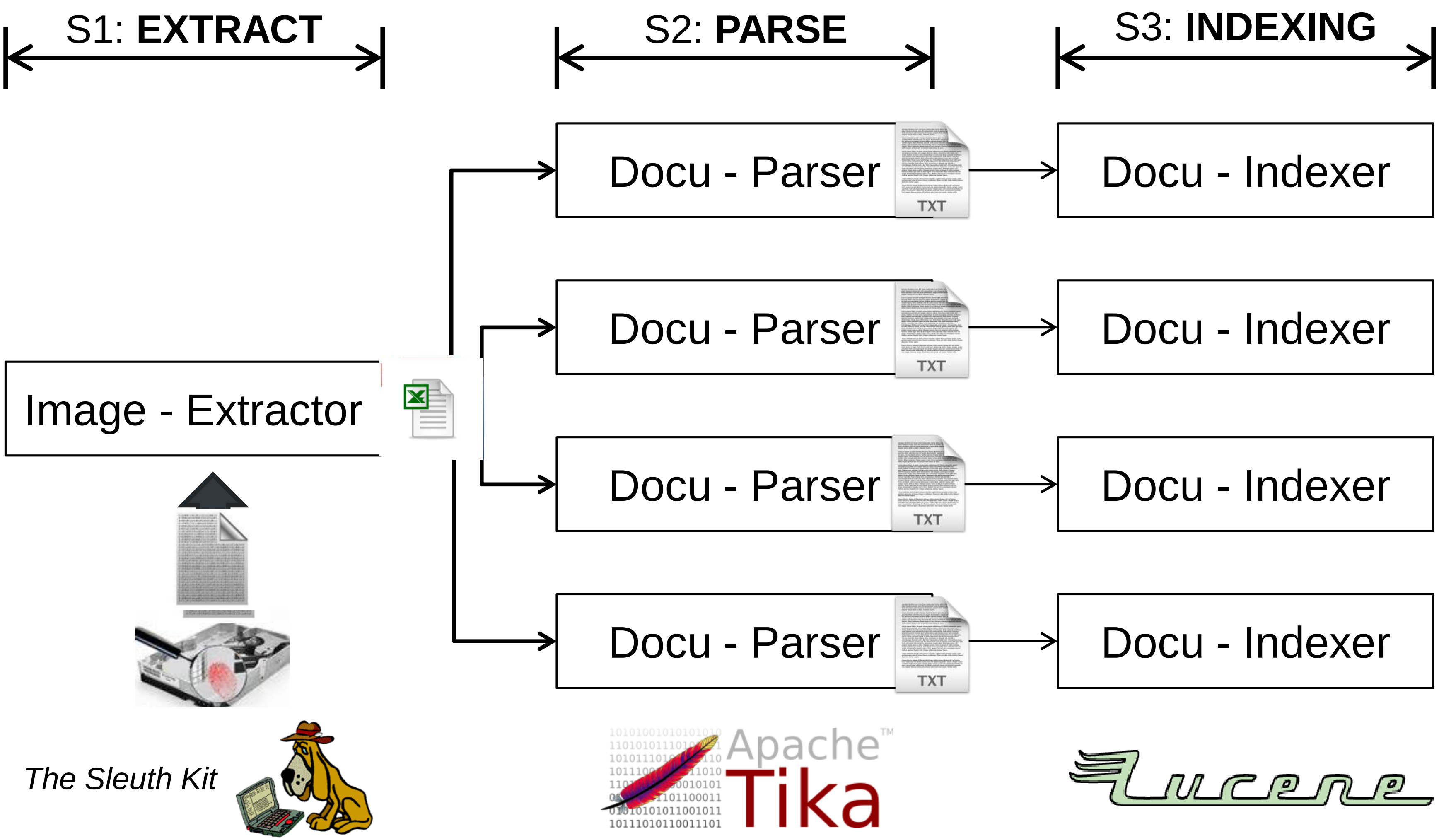
Research Statement and Hypotheses

- How to efficiently face the **indexing problem**?
  - ***Extract-Parse-Index*** in-memory pipelines
  - Parallel Technologies (e.g., *GPU*, *Pthreads*)
  - Enhancing de facto IR standard technologies
- **ISODAC** [Totaro, Bernaschi, Carbone and Cianfriglia (2016)]
  1. New high performance solution for indexing
  2. Only inverted indexes are written on disk
  3. Scalable according to hardware resources

Software Architecture



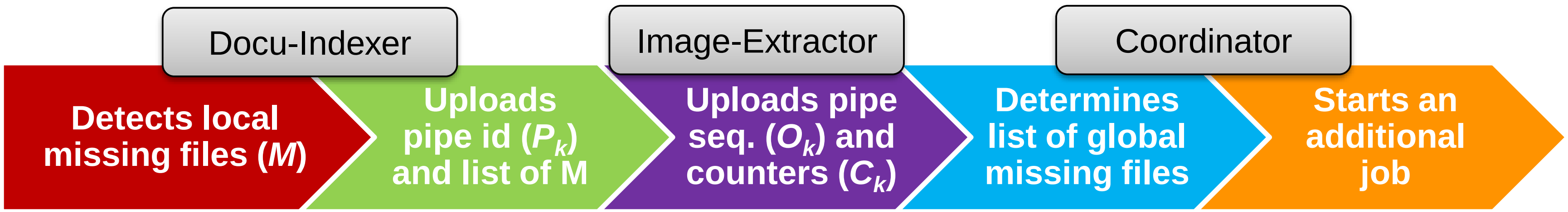
EPI In-Memory Pipeline



EPI In-Memory Pipeline

Recovery Procedure

- Provides resilience against both missing files (*M*) and parsing failures (*rounds*)



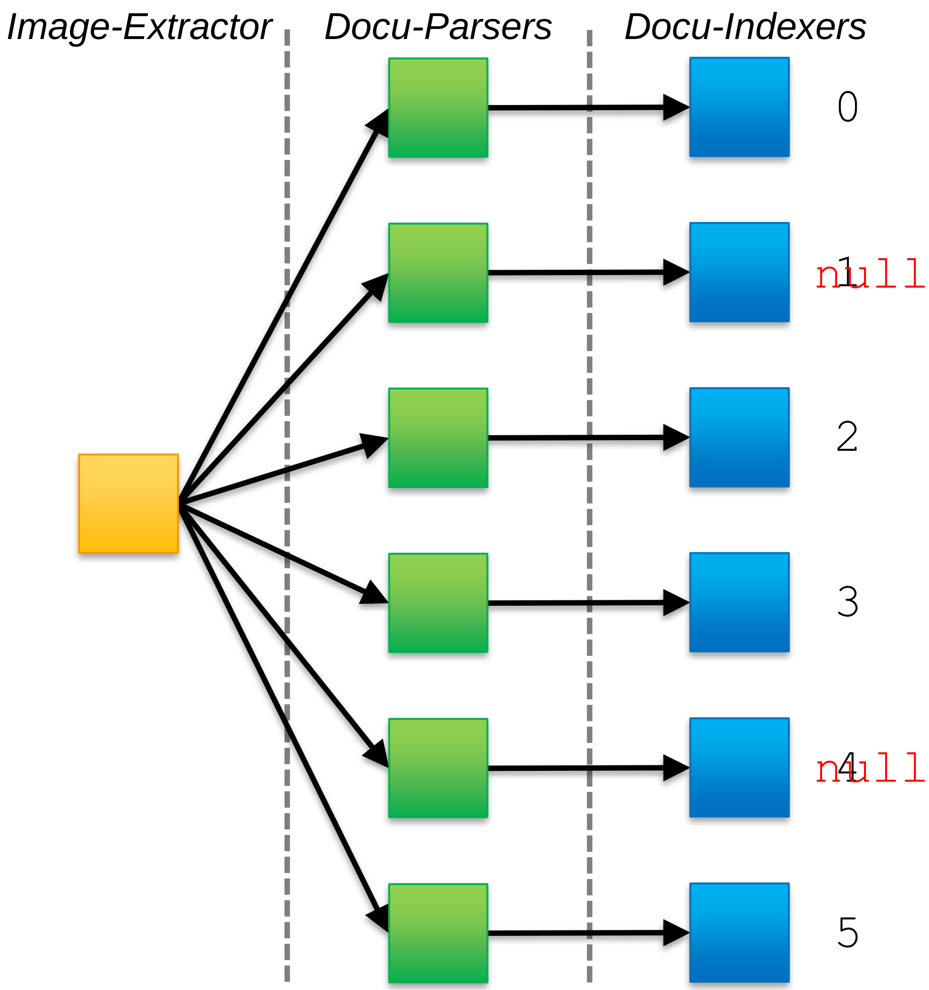
- Each global *fileID* can be calculated by using the following formula:

$$fileID = \begin{cases} M \times N[0] + O_k[0] & |rounds| = 0 \\ sent[z] + (M - C_k[z]) \times N[z+1] + O_k[z+1] & |rounds| > 0 \end{cases}$$

$$(z = argmax_i | C_k[i] < M)$$

EPI In-Memory Pipeline

Recovery Procedure



- First *round*

$i = 0$

$N[0] = 6$

$O[0] = <0,1,2,3,4,5>$

$C[0] = <x_0,x_1,x_2,x_3,x_4,x_5>$

- Second *round*

$i+1 = 1$

$N[1] = 4$

$O[1] = <0,null,1,2,null,3>$

$C[1] = <y_0,0,y_1,y_2,0,y_3>$

Optimizations: Text Analysis on GPU

- CUDA-based Text Analysis
  - Text analysis including *tokenization*, *filtering* and *stop words removal* processes
  - Exploits the computational power of ***GPU*** cards
  - Extends CLucene *StandardAnalyzer* with CUDA kernels that perform text analysis

CLucene Libraries: `http://clucene.sourceforge.net/`

Optimizations: Text Analysis on GPU

M	y		n	a	m	e		i	s		B	o	b	.	\0

One CUDA Thread per character.
Each thread applies **LowerCase Filter**

m	y		n	a	m	e		i	s		b	o	b	.	\0

Each CUDA Thread performs **Tokenization** by locating delimiter positions

-1	-1	2	-1	-1	-1	-1	7	-1	-1	10	-1	-1	-1	14	15

Vector processing in order to create two vectors representing start and end token indexes respectively.

Start Indexes (related to input text)

0	3	8	11

End Indexes (related to input text)

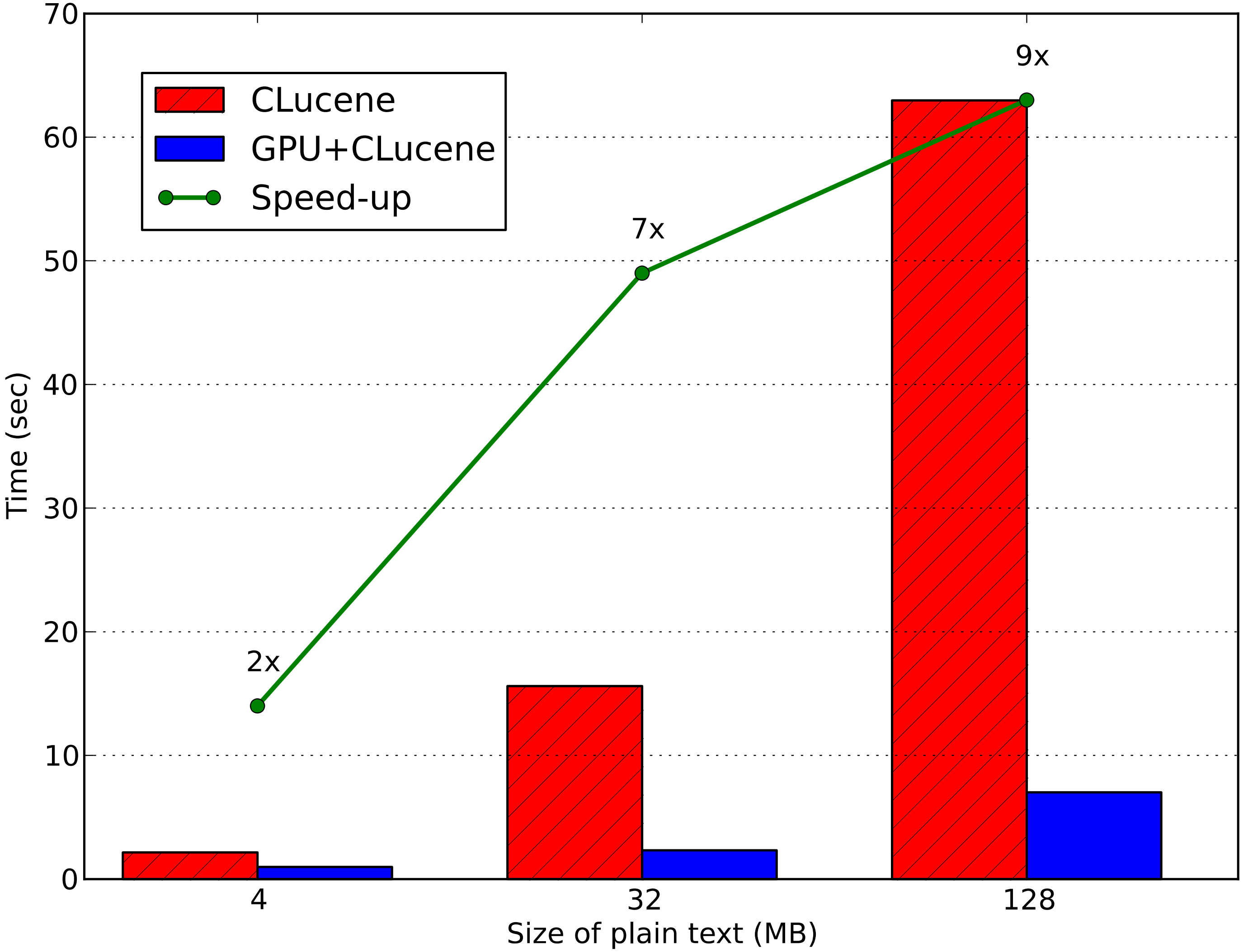
2	7	10	14

my
name
is
bob

One CUDA Thread per token.
Each thread applies **StopWords Filter.**

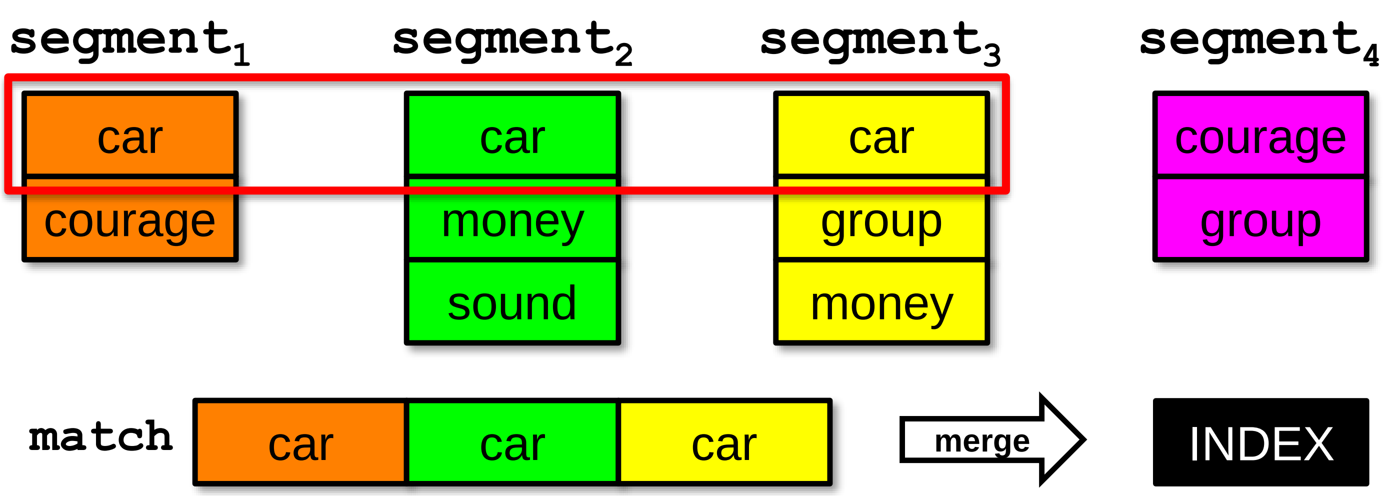
my
name
bob

Optimizations: Text Analysis on GPU



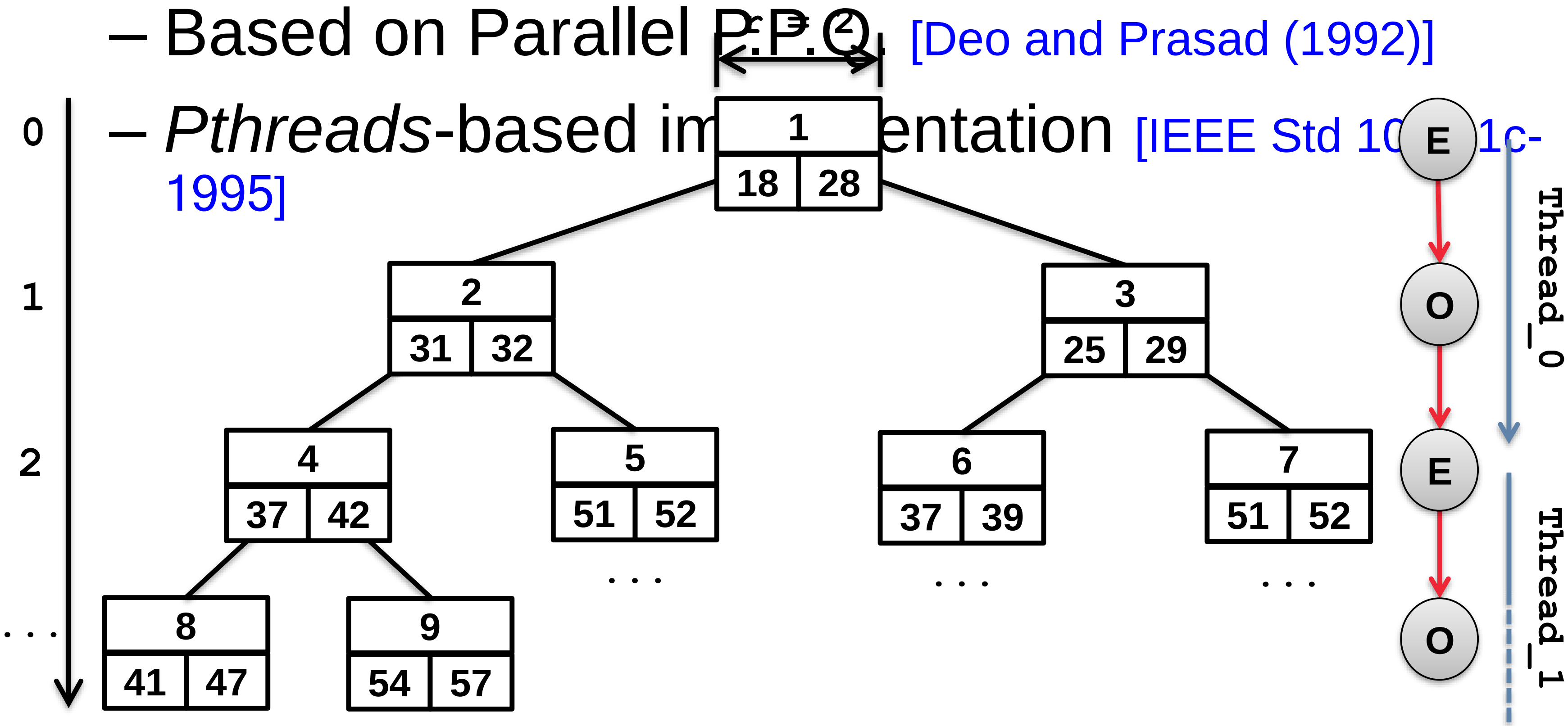
Optimizations: Parallel Segment Merging

- A Lucene index may consist of multiple sub-indexes or *segments*
- Merging algorithms are usually based on priority queues

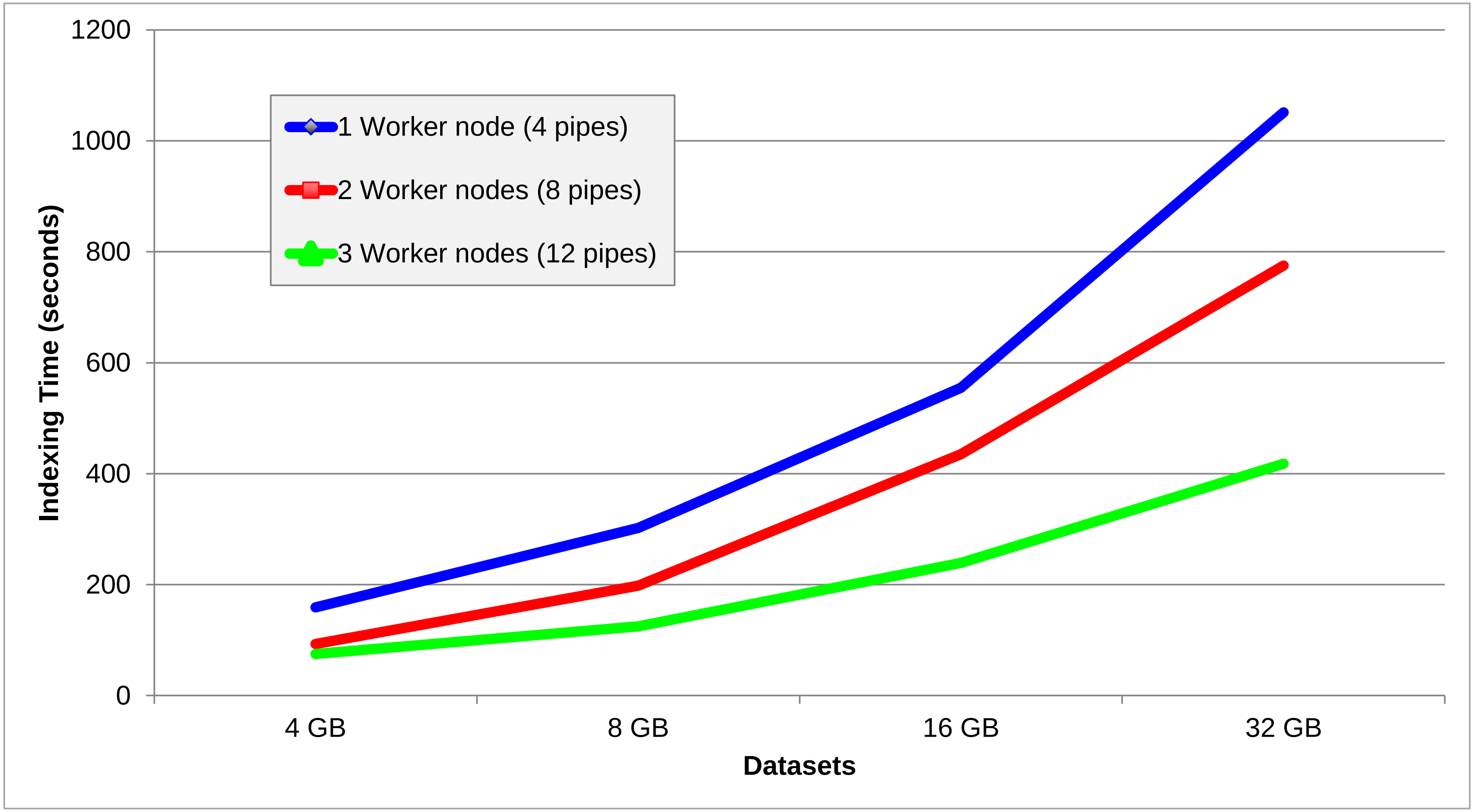


Optimizations: Parallel Segment Merging

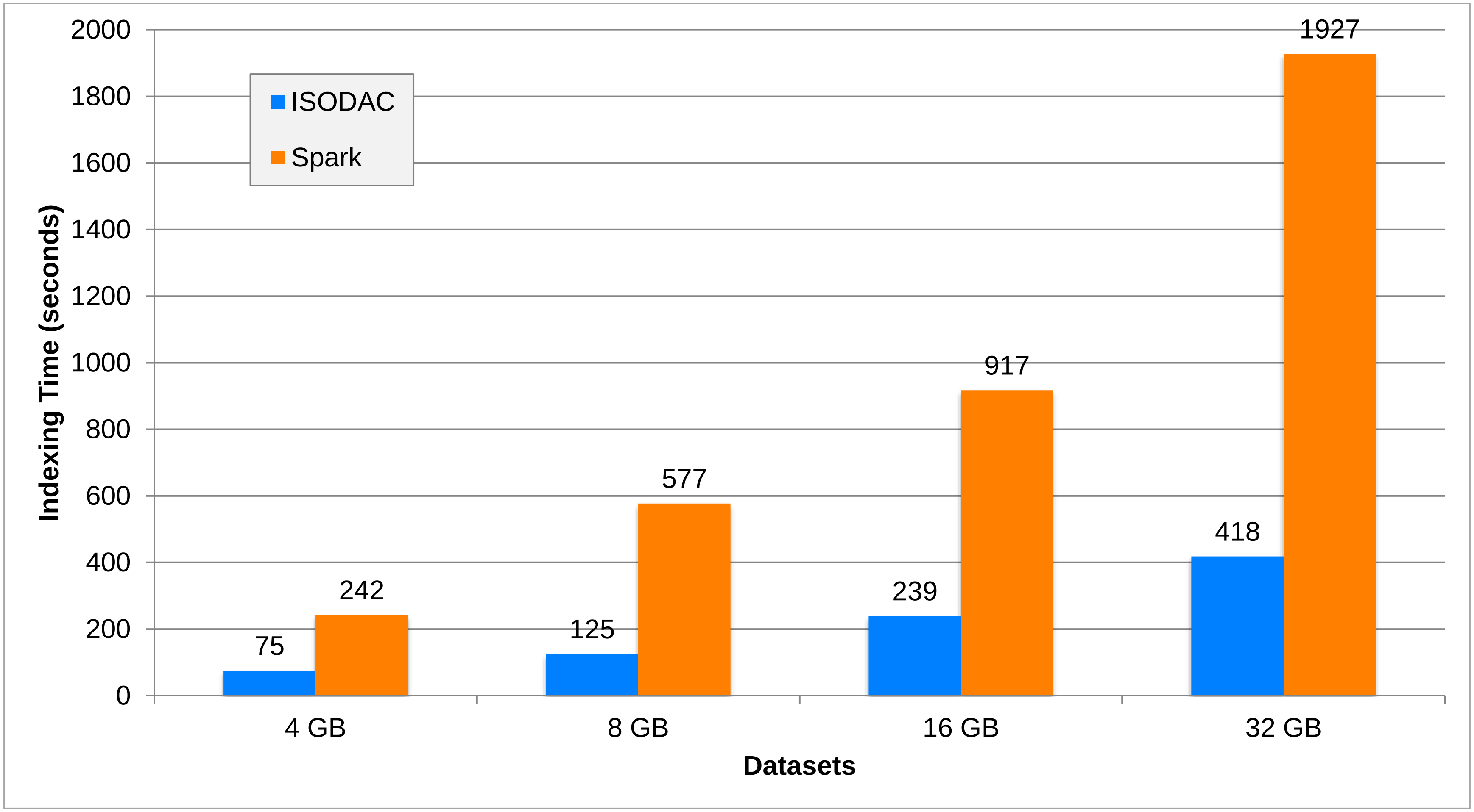
- Perform in parallel multi-thread heapify and merging of extracted data
  - Based on Parallel P.Q. [Deo and Prasad (1992)]
  - *Pthreads*-based implementation [IEEE Std 1003.1c-1995]



Evaluation: ISODAC performance



ISODAC vs. Spark Indexing



Fault Recovery

- Simulation of two kinds of failure:
  - **Transient:** node restarts after the failure
  - **Permanent:** node becomes permanently unavailable

Stress Pattern (16 GB dataset)	ISODAC		Spark	
	Time (secs)	Overhead	Time (secs)	Overhead
No Failure	239	-	917	-
Transient	387	61%	1131	45%
Permanent	415	73%	2542	177%

Metadata

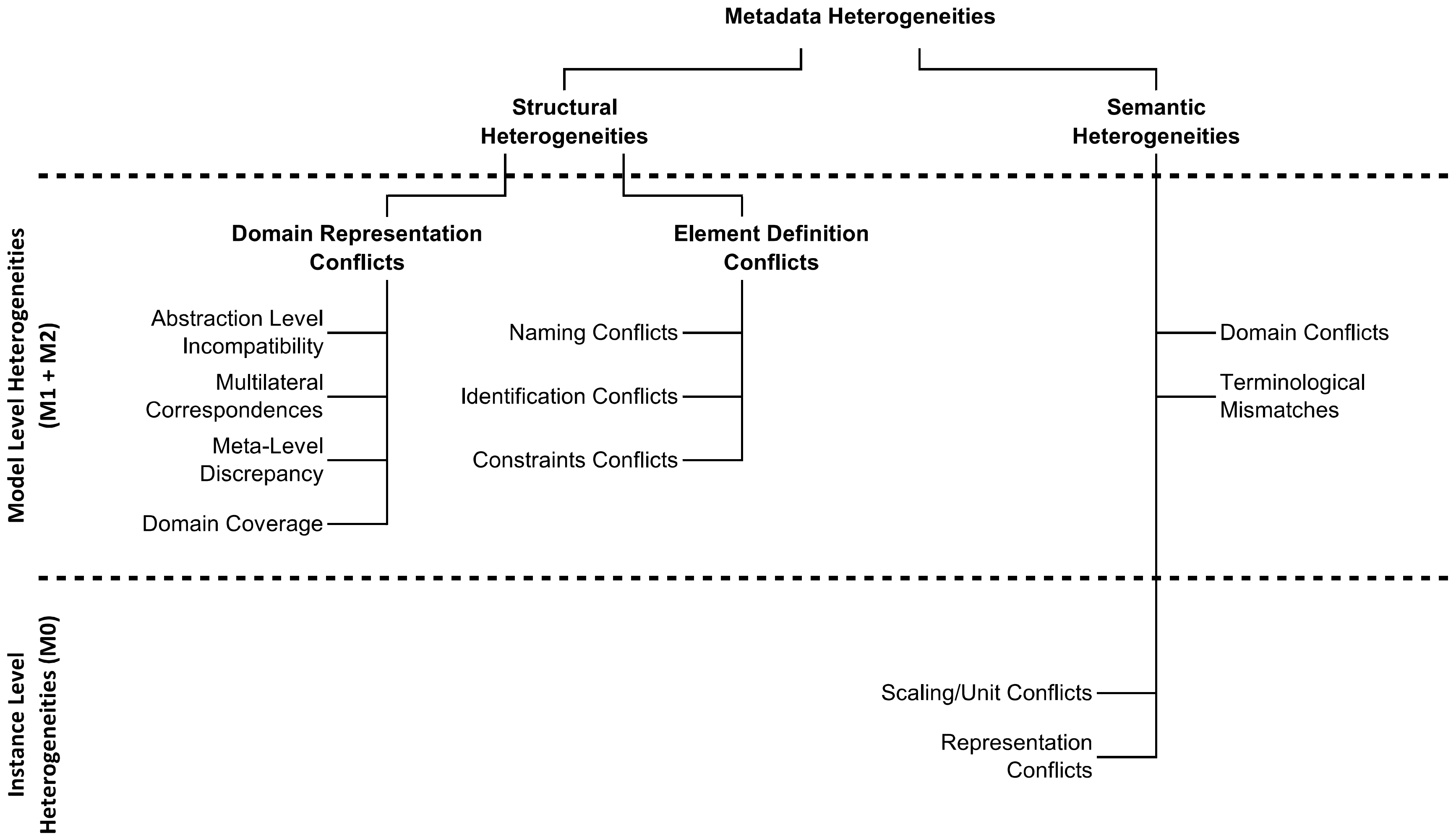
- Illustrative Example
- Metadata Building Blocks
- Metadata — A Model Perspective

Metadata Interoperability

- Prerequisite for uniform access to media objects

"*Metadata interoperability is a qualitative property of metadata information objects that enables systems and applications to work with or use these objects across system boundaries.*" [Haslhofer and Klas (2010)]

Metadata Heterogeneities



Predominant heterogeneities have been originally identified by:
[Sheth, Larson (1990)] [Ouksel, Sheth (1999)] [Wache (2003)] [Visser et al. (1997)]

Metadata Mapping

- Interoperability technique that subsumes:
  - *schema mapping* → *crosswalks*
  - *instance transformation* → *functions*

"*Given a source schema $S^s \in \mathcal{S}$ and a target schema $S^t \in \mathcal{S}$, each consisting of a set of schema elements, $e^s \in S^s$ and $e^t \in S^t$ respectively, a mapping $M \in \mathcal{M}$ is a directional relationship between a set of elements $e_i^s \in S^s$ and a set of elements $e_j^t \in S^t$.*" [Haslhofer and Klas (2010)]

Mapping Relationship

$$m \in M \begin{cases} |m| \\ p \in P \\ f \in F \end{cases}$$

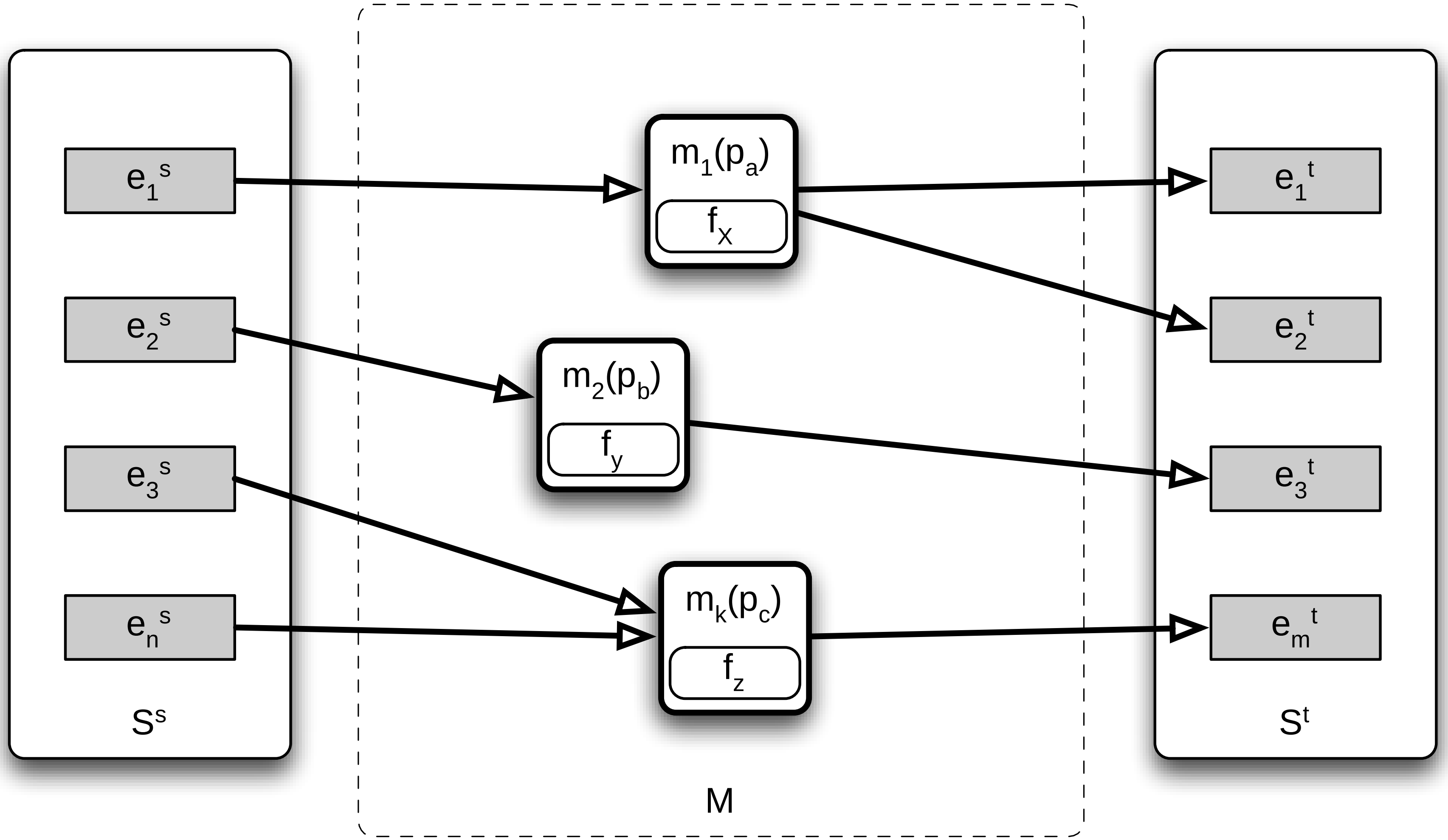
Cardinality (e.g., 1:1, 1:n, n:1)

Mapping expression

Instance transformation function

- Mapping expressions [Spaccapietra et al. (1992)]
  - Exclude: $I(e_i^s) \cap I(e_j^t) = \emptyset$
  - Equivalent: $I(e_i^s) \equiv I(e_j^t)$
  - Include: $\left(I(e_i^s) \subseteq I(e_j^t) \vee I(e_j^t) \subseteq I(e_i^s)\right)$
  - Overlap: $\left(I(e_i^s) \cap I(e_j^t) \neq \emptyset \wedge I(e_i^s) \nsubseteq I(e_j^t) \wedge I(e_j^t) \nsubseteq I(e_i^s)\right)$

Elements of Metadata Mapping



Other Group Projects at NASA JPL

Jet Propulsion Laboratory
California Institute of Technology

Six months internship at NASA JPL working for ***Computer Science for Data Intensive Applications (398M)*** *group (Mentor: Prof. Chris Mattmann)*

Apache committer and PMC member since April 2015

Celgene needs a solution for text mining in biomedical domain

DARPA Memex for fight ...st and prevention of ...n Trafficking

THE APACHE SOFTWARE FOUNDATION
apache.org

Jet Propulsion Laboratory
California Institute of Technology

jpl.nasa.gov

Giuseppe.U.Totaro@jpl.nasa.gov

https://www.researchgate.net/profile/Giuseppe_Totaro

https://github.com/giuseppetotaro

https://www.linkedin.com/in/giuseppetotaro